



Analytic Guidelines: Creating Disability Identifiers Using the Washington Group Extended Set on Functioning (WG-ES) CSPRO Syntax

Introduction

As with the WG Short Set on Functioning (WG-SS), analysis of the WG Extended Set on Functioning (WG-ES) can also produce multiple disability identifiers based on the choice of the severity threshold or cut-off. The CSPRO syntax below, however, provides for the calculation of disability identifiers using different sets of WG-ES domains using the recommended cut-off for international comparisons (described below).

For each of the disability identifiers described, the level of inclusion is at least one domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL – or – for the domains Anxiety, Depression, Pain and Fatigue, the highest level of difficulty on a four-point scale.

Each of the four disability identifiers described in this document is defined based on the choice of domains of functioning included:

WG-SS: *Short Set:* 6 domains, 6 questions.

WG-ES 1: *Extended Set:* 10 domains, 25 questions.

WG-ES 2: *Modified Extended Set* (WG-ES MINUS Pain and Fatigue): 8 domains, 20 questions.

WG-ES 3: *Short Set Enhanced* (WG-SS PLUS Upper body, Anxiety and Depression): 8 domains, 12 questions.

NOTE: For data analysis, use your standard weighting and estimation techniques.

The CSPRO syntax is based on the *variable labels* indicated in the table below. The complete WG-ES module includes more questions than appear in this table. Disability status is determined through difficulty in the basic, universal activities *without* the use of assistive technology or other assistance. There are several mobility questions, for example, that reference difficulty walking *with* the use of assistance. Those questions are not included in the analytic plan provided here; however, they can be

The Washington Group Implementation Documents

cover the tools developed by the Washington Group on Disability Statistics (WG) to collect internationally comparable disability data on censuses and surveys. The documents address best practices in implementing the Short Set, Extended Set, Short Set – Enhanced, the WG / UNICEF Child Functioning Modules for children 2-4 and 5-17 years of age, and the WG / ILO LFS Disability Module, as well as other WG tools. Topics include translation, question specifications, analytic guidelines, programming code for analyses, the use of the tools for the purposes of disaggregation, and more.

To locate other WG Implementation Documents and more information, visit the Washington Group website:
<http://www.washingtongroup-disability.com/>.

used in other analyses that look more closely into the effect of assistive technology (environmental facilitators) on functioning.

Only those questions/variables below are used in the determination of disability identifiers.
Ensure that you use the same *variable labels* OR revise the CSPRO syntax to reflect the *variable labels* in your database.

The WG-SS is administered as part of the U.S. National Health Interview Survey (NHIS). The data used to prepare these guidelines come from the 2013 NHIS.

Note to users of the U.S. NHIS: the variable names in the NHIS data file and documentation may differ from those used in this document; e.g., the self-care domain variable referenced as SC-SS in this document is referred to as UB_SS in the NHIS data file and documentation.

The CSPRO code used to produce the outputs in this document is included in its entirety in Appendix 1.

WG Extended Set Questions	Variable Label	Response Pattern
VISION		
1. Do you have difficulty seeing even if wearing glasses?	VIS_SS	1
COMMUNICATION		
2. Using your usual language, do you have difficulty communicating (for example understanding or being understood by others)?	COM_SS	1
HEARING		
3. Do you have difficulty hearing even if using a hearing aid?	HEAR_SS	1
4. Difficulty hearing conversation with one person in quiet room?	HEAR_3	1
5. Difficulty hearing conversation with one person in noisier room?	HEAR_4	1
COGNITION		
6. Do you have difficulty remembering or concentrating?	COG_SS	1
7. Difficulty remembering, concentrating, or both?	COG_1	2
8. How often have difficulty remembering?	COG_2	3
9. Amount of things you have difficulty remembering?	COG_3	4
SELF-CARE		
10. Do you have difficulty with (self-care such as) washing all over or dressing?	SC_SS	1
UPPER BODY		

11. Difficulty raising 2 liter bottle of water from waist to eye level?	UB_1	1
12. Degree of difficulty using hands and fingers	UB_2	1
MOBILITY		
13. Do you have difficulty walking or climbing stairs?	MOB_SS	1
14. Difficulty walking 100 yards on level ground without aid or equipment?	MOB_4	1
15. Difficulty walking 1/3rd mile on level ground without aid or equipment	MOB_5	1
16. Difficulty walking up or down 12 steps without aid or equipment?	MOB_6	1
AFFECT (ANXIETY)		
17. How often feel worried, nervous, or anxious?	ANX_1	5
18. Level of feelings when last felt worried, nervous, or anxious?	ANX_3	6
AFFECT (DEPRESSION)		
19. How often do you feel depressed?	DEP_1	5
20. How depressed you felt last time you were depressed?	DEP_3	6
PAIN		
21. Frequency of pain in past 3 months?	PAIN_2	7
22. How much pain you had last time you had pain?	PAIN_4	6
FATIGUE		
23. How often felt very tired or exhausted in past 3 months?	TIRED_1	7
24. How long most recent tired or exhausted feelings lasted?	TIRED_2	8
25. Level of tiredness last time felt very tired or exhausted?	TIRED_3	6

NOTE: **Red** refers to the Washington Group Short Set (**WG-SS**).
All 25 questions are included in **WG-ES 1**.

Red plus **Blue** plus **Green** questions are included in **WG-ES 2**.

Red plus **Green** questions are included in **WG-ES 3**.

Response patterns:

	Pattern 1	Pattern 2	Pattern 3	Pattern 4
1	No difficulty	Difficulty remembering only	Sometimes	A few things
2	Yes, Some difficulty	Difficulty concentrating only	Often	A lot of things
3	Yes, A lot of difficulty	Difficulty both remembering & concentrating	All of the time	Almost everything
4	Cannot do at all			
7	Refused			
8	Not ascertained			
9	Don't know			

	Pattern 5	Pattern 6*	Pattern 7	Pattern 8
1	Daily	A little	Never	Some of the day
2	Weekly	A lot	Some days	Most of the day
3	Monthly	Somewhere in between a little and a lot	Most days	All of the day
4	A few times a year		Every day	
5	Never			
7	Refused			
8	Not ascertained			
9	Don't know			

*** IN THE SYNTAX BELOW, NOTE THAT ITEMS WITH RESPONSE PATTERN 6 (ANX_3, DEP_3, PAIN_4 AND TIRED_3) ARE RECODED TO PLACE “SOMEWHERE BETWEEN” NUMERICALLY IN-BETWEEN “A LITTLE” AND “A LOT”.**

The WG-SS is embedded within the WG-ES.

The WG-ES is supplemented with:

- additional questions to those existing 6 domains and
- additional domains (several with multiple questions).

The CSPRO syntax presented below includes a couple of elements that were particular to the content of the WG-ES.

First, it was important to determine single domain-specific identifiers for those domains of functioning that included multiple questions. For example, upper body functioning includes two questions, each eliciting specific and unique actions: difficulty raising a bottle of water from waist to eye level (arms/shoulders), and difficulty using hands and fingers. Those two questions were analyzed and combined to produce a single upper body indicator with four levels of difficulty ranging from 1 - low

difficulty to 4 - high difficulty – not unlike the categorical responses to the single WG-SS questions: no difficulty, some difficulty, a lot of difficulty and cannot do at all. As with the upper body domain, the WG-ES domains cognition, anxiety, depression, pain and fatigue have different response patterns that do not readily ‘translate’ into the usual WG response pattern. For these domains of functioning, a similar 4-scale response pattern was produced and annotated as level 1 through 4, where 1 is the lowest level of difficulty and 4 is the highest.

Second, individual domain indicators were assessed together to determine the appropriate cut-off for inclusion into an overall disability identifier – for the purposes of estimating prevalence and disaggregating outcome indicators by disability status.

NOTE:

For all variables, codes (7) *Refused*, (8) *Not Ascertained*, and (9) *Don't know*, are recoded to *Missing*.

CSPRO WG Extended Set Syntax Annotated with Output Tables

Actual CSPRO syntax is indented and are in **Blue text**.

NOTE: For data analysis, use your standard weighting and estimation techniques.

NOTE: Frequency distributions are generated using the "Tabulate Frequencies" option in CPro (https://www.csprouters.org/help/CSFreq/introduction_to_tabulate_frequencies.html)

This will be noted in the syntax below with: **Tabulate Frequencies**.

NOTE: : Cross tabulations are generated using the "Create a Tabulation Application " option in CPro (https://www.csprouters.org/help/GetStart/exercise_05_01_create_a_tabulation_application.html)

This will be noted in the syntax below with: **Cross Tabulations**.

VISION

Step 1. Generate frequency distribution for Vision domain.

VIS_SS is the WG-SS Vision question.

```
PROC VISION  
IF VIS_SS IN 1, 2, 3, 4 THEN Vision=VIS_SS;  
ELSEIF VIS_SS IN 7, 8, 9 THEN Vision=NotAppl;  
ENDIF;
```

Tabulate Frequencies.

Vision: Degree of difficulty seeing

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13690	79.0	81.6	81.6
	Some difficulty	2708	15.6	16.2	97.8
	A lot of difficulty	333	1.9	2.0	99.8
	Cannot do at all	36	.2	.2	100.0
	Total	16767	96.8	100.0	
Missing		559	3.2		
Total		17326	100.0		

COMMUNICATION

Step 2. Generate frequency distribution for Communication domain.

COM_SS is the WG-SS Communication question.

PROC COMMUNICATION

If COM_SS in 1, 2, 3,4 **then** Communication =COM_SS;

ElseIf COM_SS in 7, 8, 9 **then** Communication = **NotAppl**;

EndIf;

Tabulate Frequencies.

Communication: Degree of difficulty communicating using usual language

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15874	91.6	94.7	94.7
	Some difficulty	745	4.3	4.4	99.2
	A lot of difficulty	94	.5	.6	99.7
	Cannot do at all	43	.2	.3	100.0
	Total	16756	96.7	100.0	
Missing		570	3.3		
Total		17326	100.0		

HEARING

Step 3. Generate frequency distributions and cross-tabulations for Hearing domain questions and determine Hearing Indicator

HEAR_SS is the WG-SS Hearing question.

PROC HEARING

If HEAR_SS **in** 1, 2, 3, 4 **then** Hearing =HEAR_SS;
ElseIf HEAR_SS **in** 7, 8, 9 **then** Hearing = **NotAppl**;
EndIf;

Tabulate Frequencies.

HEAR_3 is *Difficulty hearing conversation with one person in quiet room.*

If HEAR_3 **in** 1, 2, 3, 4 **then** HEAR_3_R=HEAR_3;
Else If HEAR_3 **in** 7, 8, 9 **then** HEAR_3_R= **NotAppl**;
EndIf;

Tabulate Frequencies.

HEAR_4 is *Difficulty hearing one person in noisier room.*

If HEAR_4 **in** 1, 2, 3, 4 **then** HEAR_4_R=HEAR_4;
Else If HEAR_4 **in** 7, 8, 9 **then** HEAR_4_R= **NotAppl**;
EndIf;

Tabulate Frequencies.

Hearing: Degree of difficulty hearing

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13680	79.0	81.6	81.6
	Some difficulty	2753	15.9	16.4	98.0
	A lot of difficulty	310	1.8	1.8	99.9
	Cannot do at all	23	.1	.1	100.0
	Total	16766	96.8	100.0	
Missing		560	3.2		
Total		17326	100.0		

HEAR_3_R : Difficulty hearing conversation with one person in quiet room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15249	88.0	91.0	91.0
	Some difficulty	1316	7.6	7.9	98.9
	A lot of difficulty	162	.9	1.0	99.9
	Cannot do at all	10	.1	.1	100.0
	Total	16737	96.6	100.0	
Missing		589	3.4		
Total		17326	100.0		

HEAR_4_R: Difficulty hearing one person in noisier room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	11697	67.5	70.0	70.0
	Some difficulty	4191	24.2	25.1	95.1
	A lot of difficulty	779	4.5	4.7	99.7
	Cannot do at all	48	.3	.3	100.0
	Total	16715	96.5	100.0	
Missing		611	3.6		
Total		17326	100.0		

Step 4. For Hearing questions, recode HEAR_3_R and HEAR_4_R to value 4 (cannot do at all) if Hear_SS is 4 (Cannot do at all).

The syntax below recodes HEAR_3_R and HEAR_4_R to 4 (cannot do at all) if Hear_SS is 4 (cannot do at all).

```
PROC HEAR_3_R
```

```
If Hearing = 4 and HEAR_3_R = NotAppl then HEAR_3_X = 4;
```

```
Else HEAR_3_X=HEAR_3_R;
```

```
EndIf;
```

Tabulate Frequencies.

```
PROC HEAR_4_R
```

```
If Hearing = 4 and HEAR_4_R = NotAppl then HEAR_4_X = 4;
```

```
Else HEAR_4_X=HEAR_4_R;
```

```
EndIf;
```

Tabulate Frequencies.

HEAR_3_X : Difficulty hearing conversation with one person in quiet room

		Frequency	Per cent	Valid Percent	Cumulative Percent
Valid	No difficulty	15249	88.0	91.0	91.0
	Some difficulty	1316	7.6	7.9	98.8
	A lot of difficulty	162	.9	1.0	99.8
	Cannot do at all	33	.2	.2	100.0
	Total	16760	96.7	100.0	
Missing		566	3.3		
Total		17326	100.0		

HEAR_4_X: Difficulty hearing one person in noisier room

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	11697	67.5	69.9	69.9
	Some difficulty	4191	24.2	25.0	94.9
	A lot of difficulty	779	4.5	4.7	99.6
	Cannot do at all	71	.4	.4	100.0
	Total	16738	96.6	100.0	
Missing		588	3.4		
Total		17326	100.0		

Step 5. Generate a cross-tabulation of the two Hearing Extended Set questions: HEAR_3_X and HEAR_4_X.

The syntax below produces a cross-tabulation of the two Extended Set questions: HEAR_3_X: *Difficulty hearing in a quiet room* and HEAR_4_X: *Difficulty hearing in a noisier room* to determine a single HEARING INDICATOR.

Cross Tabulations.

HEAR_4_X (Difficulty hearing in a Noisier room)		HEAR_3_X : Difficulty hearing conversation with one person in quiet room				Total
		No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	
No difficulty		11603	94	0	0	11697
Some difficulty		3373	809	8	0	4190
A lot of difficulty		253	388	138	0	779
Cannot do at all		8	24	16	23	71
Total		15237	1315	162	23	16737

Step 6. Create a HEARING INDICATOR (H_INDICATOR) based on the two additional hearing questions HEAR_3_X and HEAR_4_X.

The syntax below creates a HEARING INDICATOR (H_INDICATOR) based on the cross-tabulation of the two additional hearing questions HEAR_3_X and HEAR_4_X.

```

PROC H_INDICATOR
if (HEAR_3_X = 1 and HEAR_4_X = 1) or (HEAR_3_X = 1 and HEAR_4_X = 2)
then H_INDICATOR = 1;
elseif (HEAR_3_X = 2 and (HEAR_4_X = 1 or HEAR_4_X = 2)) or (HEAR_3_X = 1 and HEAR_4_X = 3) then H_INDICATOR = 2;
elseif (HEAR_3_X = 3 and (HEAR_4_X = 1 or HEAR_4_X = 2) or (HEAR_3_X = 2 and HEAR_4_X = 3) or (HEAR_3_X = 1 and HEAR_4_X = 4)) then H_INDICATOR = 3;

```

```

elseif ((HEAR_3_X = 3 and HEAR_4_X = 3) or HEAR_3_X = 4 or (HEAR_4_X = 4 and
(HEAR_3_X = 2 or HEAR_3_X = 3))) then H_INDICATOR = 4;
endif;

```

Tabulate Frequencies.

		H_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	14976	86.4	89.4	89.4
	2.00	1156	6.7	6.9	96.3
	3.00	404	2.3	2.4	98.7
	4.00	211	1.2	1.3	100.0
	Total	16747	96.7	100.0	
Missing		579	3.3		
Total		17326	100.0		

COGNITION: Degree of difficulty remembering or concentrating

Step 7. Generate frequency distributions and cross-tabulations for Cognition domain questions and determine a Cognition Indicator.

COG_SS is the WG-SS Cognition question.

```

PROC COGNITION
If COG_SS in 1, 2, 3, 4 then COGNITION =COG_SS;
Elseif COG_SS in 7, 8, 9 then COGNITION = NotAppl;
Endif;

```

Tabulate Frequencies.

Cognition: Degree of difficulty remembering or concentrating

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13719	79.2	81.9	81.9
	Some difficulty	2632	15.2	15.7	97.6
	A lot of difficulty	382	2.2	2.3	99.9
	Cannot do at all	20	.1	.1	100.0
	Total	16753	96.7	100.0	
Missing		573	3.3		
Total		17326	100.0		

If response to COG-SS is *some difficulty*, *a lot of difficulty* or *cannot do at all*, then the respondent is asked COG_1: whether they have difficulty remembering, concentrating or both.

```

PROC COG_1_R

```

If COG_1 **in** 1, 2, 3 **then** COG_1_R=COG_1;
ElseIf COG_1 **in** 7, 8, 9 **then** COG_1_R= **NotAppl**;
EndIf;

Tabulate Frequencies.

COG_1_R: Difficulty remembering, concentrating, or both?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Difficulty remembering only	983	5.7	32.4	32.4
	Difficulty concentrating only	388	2.2	12.8	45.2
	Difficulty with both remembering and concentrating	1659	9.6	54.8	100.0
	Total	3030	17.5	100.0	
Missing		14296	82.5		
Total		17326	100.0		

Step 8. Account for those who did not answer COG_1 (COG_SS is 1 – no difficulty and they were skipped) by recoding COG_1 to 0 (No difficulty).

If response to COG-SS is 1: no difficulty, then the variable COG_1_R is recoded into COG_1A, and the value assigned is 0: no difficulty.

PROC COG_1A
If COG_SS=1 **then** COG_1A=0;
Else COG_1A=COG_1_R;
EndIf;

Tabulate Frequencies.

COG_1A: Difficulty remembering, concentrating, or both? (including no difficulty)

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13719	79.2	81.9	81.9
	Difficulty remembering only	983	5.7	5.9	87.8
	Difficulty concentrating only	388	2.2	2.3	90.1
	Difficulty with both remembering and concentrating	1659	9.6	9.9	100.0
	Total	16749	96.7	100.0	
Missing		577	3.3		
Total		17326	100.0		

Step 9. Generate frequency distribution for remaining cognition questions.

Frequency distribution of the Cognition extended REMEMBERING questions: COG_2 *How often have difficulty remembering*, and COG_3 *Amount of things you have difficulty remembering*.

```
PROC COG_2_R
If COG_2 in 1, 2, 3 then COG_2_R=COG_2;
ElseIf COG_2 in 7, 8, 9 then COG_2_R= NotAppl;
EndIf;
```

Tabulate Frequencies.

```
PROC COG_3_R
If COG_3 in 1, 2, 3 then COG_3_R=COG_3;
ElseIf COG_3 in 7, 8, 9 then COG_3_R= NotAppl;
EndIf;
```

Tabulate Frequencies.

COG_2_R: How often have difficulty remembering?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Sometimes	1916	11.1	72.4	72.4
	Often	513	3.0	19.4	91.8
	All of the time	216	1.2	8.2	100.0
	Total	2645	15.3	100.0	
Missing		14681	84.7		
Total		17326	100.0		

COG_3_R: Amount of things you have difficulty remembering?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	A few things	2119	12.2	80.3	80.3
	A lot of things	386	2.2	14.6	94.9
	Almost everything	134	.8	5.1	100.0
	Total	2639	15.2	100.0	
Missing		14687	84.8		
Total		17326	100.0		

Step 10. Generate cross-tabulation of the two Cognition extended set questions COG_2R by COG_3_R.

The syntax below produces a cross-tabulation of the two Extended Set REMEMBERING questions: COG_2_R: *How often you have difficulty remembering* and COG_3_R: *The amount of things you have difficulty remembering* to determine a single REMEMBERING INDICATOR.

Cross Tabulations.

COG_2_R: How often do you have difficulty remembering?		COG_3_R: Amount of things you have difficulty remembering?			Total
		A few things	A lot of things	Almost everything	
How often have difficulty remembering?	Sometimes	1788	105	20	1913
	Often	279	197	34	510
	All of the time	51	84	80	215
Total		2118	386	134	2638

Step 11. Create a Remembering Indicator based on distribution of COG_2_R and COG_3_R.

The syntax below creates a REMEMBERING INDICATOR (R_INDICATOR) based on the two additional remembering questions (COG_2_R and COG_3_R).
 If Cognition is 1: no difficulty, then the Remembering Indicator is coded as 1: the lowest level of difficulty.

```

PROC R_INDICATOR
if COGNITION = 1 then R_INDICATOR = 1;
elseif ((COG_2_R = 1 and COG_3_R = 1) or (COG_3_R = 1 and COG_2_R = 2) or
(COG_3_R = 2 and COG_2_R = 1)) then R_INDICATOR = 2;
elseif (COG_3_R = 2 and COG_2_R = 2) then R_INDICATOR = 3;
elseif (COG_3_R = 3 or COG_2_R = 3) then R_INDICATOR = 4;
    
```

Step 12. If COG_1A is coded as 2 (concentrating only), then the Remembering Indicator is coded as 5.

These 388 individuals are respondents who were not included in the Remembering Indicator since they had only difficulty concentrating.

```

elseif (COG_1A = 2) then R_INDICATOR = 5;
else R_INDICATOR = 0;
endif;
    
```

Tabulate Frequencies.

Step 13. Generate frequency distribution of the Remembering Indicator.

		R_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	.00	580	3.3	3.3	3.3
	1.00	13719	79.2	79.2	82.5
	2.00	2172	12.5	12.5	95.1
	3.00	197	1.1	1.1	96.2
	4.00	270	1.6	1.6	97.8
	5.00	388	2.2	2.2	100.0
Total		17326	100.0	100.0	

Step 14. Supplement Remembering Indicator with information on difficulty concentrating.

The syntax below adds information on whether cognitive difficulties are compounded by difficulty concentrating in addition to difficulty remembering.

Create a COGNITION INDICATOR (COG_INDICATOR) based on R_INDICATOR (above) and the cognition question (COG_1_R).

The 388 individuals with ‘concentrating only’ were allocated as follows:

1. 357 with a little difficulty on Cognition question were classified as **2**
2. 30 with a lot of difficulty on Cognition question were classified as **3**
3. 1 with cannot do on Cognition question was classified as **4**

Those with both remembering and concentrating difficulty were upgraded 36 individuals from **2** to **3**, and 125 individuals from **3** to **4**.

PROC COG_INDICATOR

COG_INDICATOR = R_INDICATOR;

if R_INDICATOR = **5** **and** COGNITION = **2** **then** COG_INDICATOR = **2**;

elseif R_INDICATOR = **5** **and** COGNITION = **3** **then** COG_INDICATOR = **3**;

elseif R_INDICATOR = **5** **and** COGNITION = **4** **then** COG_INDICATOR = **4**;

endif;

if R_INDICATOR = **2** **and** COG_1_R = **3** **and** COGNITION = **3** **then** COG_INDICATOR = **3**;

elseif R_INDICATOR = **3** **and** COG_1_R = **3** **and** COGNITION = **3** **then** COG_INDICATOR = **4**;

endif;

Tabulate Frequencies.

Step 15. Generate frequency distribution of the Cognition Indicator.

		COG_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	.00	580	3.3	3.3	3.3
	1.00	13719	79.2	79.2	82.5
	2.00	2449	14.1	14.1	96.7
	3.00	226	1.3	1.3	98.0
	4.00	352	2.0	2.0	100.0
Total		17326	100.0	100.0	

UPPER BODY

Step 16. Generate frequency distributions and cross-tabulations for Self-care and Upper body domain questions and determine the Upper Body Indicator.

SC_SS is the WG-SS Self-care question.

UB_1 is Difficulty raising 2 liter bottle of water from waist to eye level.

UB_2 is Difficulty using hands and fingers

```
PROC Self_Care
If SC_SS in 1, 2, 3, 4 then Self_Care=SC_SS;
ElseIf SC_SS in 7, 8, 9 then Self_Care= NotAppl;
EndIf;
```

Tabulate Frequencies.

```
If UB_1 in 1, 2, 3, 4 then UB_1_R=UB_1;
ElseIf UB_1 in 7, 8, 9 then UB_1_R= NotAppl;
EndIf;
```

Tabulate Frequencies.

```
If UB_2 in 1, 2, 3, 4 then UB_2_R=UB_2;
ElseIf UB_2 in 7, 8, 9 then UB_2_R= NotAppl;
EndIf;
```

Tabulate Frequencies.

First, calculate frequency distributions on the short set and two extended set questions.

SELF_CARE: Degree of difficulty with self-care

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	16029	92.5	95.7	95.7
	Some difficulty	544	3.1	3.2	98.9
	A lot of difficulty	114	.7	.7	99.6
	Cannot do at all	68	.4	.4	100.0
	Total	16755	96.7	100.0	
Missing		571	3.3		
Total		17326	100.0		

UB_1_R: Diff raising 2 liter bottle of water from waist to eye level

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15677	90.5	93.6	93.6
	Some difficulty	743	4.3	4.4	98.0
	A lot of difficulty	167	1.0	1.0	99.0
	Cannot do at all	166	1.0	1.0	100.0
	Total	16753	96.7	100.0	
Missing		573	3.3		
Total		17326	100.0		

UB_2_R: Degree of difficulty using hands and fingers

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	15199	87.7	90.7	90.7
	Some difficulty	1229	7.1	7.3	98.1
	A lot of difficulty	255	1.5	1.5	99.6
	Cannot do at all	70	.4	.4	100.0
	Total	16753	96.7	100.0	
Missing		573	3.3		
Total		17326	100.0		

Step 17. Generate a cross-tabulation of the two Upper body Extended Set questions: UB_2_R and UB_1_R.

The syntax below produces a cross-tabulation of the two Extended Set questions: UB_1_R: *Difficulty raising a 2 liter bottle of water from waste to eye level* UB_2_R: *Difficulty using hands and fingers* to determine a single UPPER BODY INDICATOR (UB_INDICATOR).

Cross Tabulations.

UB_1_R: Diff raising 2 liter bottle of water from waist to eye level

UB_2_R: Difficulty using hands and fingers		UB_1_R: Diff raising 2 liter bottle of water from waist to eye level				Total
		No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	
Degree of difficulty using hands and fingers	No difficulty	14786	309	58	44	15197
	Some difficulty	782	355	51	40	1228
	A lot of difficulty	98	73	51	33	255
	Cannot do at all	9	5	7	49	70
Total		15675	742	167	166	16750

Step 18. Create an UPPER BODY INDICATOR (UB_INDICATOR) based on the two additional upper body questions UB_2_R and UB_3_R.

Syntax below creates UB_INDICATOR based on the distribution in the cross-tabulation above.

```
PROC UB_INDICATOR
if UB_1_R = 4 or UB_2_R = 4 then UB_INDICATOR = 4;
elseif UB_INDICATOR <> 4 and (UB_1_R = 3 or UB_2_R = 3) then UB_INDICATOR = 3;
elseif UB_INDICATOR <> 4 and UB_INDICATOR <> 3 and (UB_1_R = 2 or UB_2_R = 2)
then UB_INDICATOR = 2;
elseif UB_INDICATOR <> 4 and UB_INDICATOR <> 3 and UB_INDICATOR <> 2 and
(UB_1_R = 1 or UB_2_R = 1) then UB_INDICATOR = 1;
endif;
```

Tabulate Frequencies.

		UB_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	14790	85.4	88.3	88.3
	2.00	1448	8.4	8.6	96.9
	3.00	331	1.9	2.0	98.9
	4.00	187	1.1	1.1	100.0
	Total	16756	96.7	100.0	
Missing		570	3.3		
Total		17326	100.0		

MOBILITY

Step 19. Generate frequency distributions and cross-tabulations for Mobility domain questions and determine Mobility Indicator.

MOB_SS is the WG-SS Mobility question.

MOB_4 is Difficulty walking 100 yards on level ground without aid or equipment.
 MOB_5 is Difficulty walking 1/3rd mile on level ground without aid or equipment.

```
PROC MOBILITY
If MOB_SS in 1, 2, 3, 4 then Mobility=MOB_SS;
ElseIf MOB_SS in 7, 8, 9 then Mobility= NotAppl;
EndIf;
```

Tabulate Frequencies.

```
If MOB_4 in 1, 2, 3, 4 then MOB_4_R=MOB_4;
ElseIf MOB_4 in 7, 8, 9 then MOB_4_R= NotAppl;
```

Tabulate Frequencies.

```
If MOB_5 in 1, 2, 3, 4 then MOB_5_R=MOB_5;
ElseIf MOB_5 in 7, 8, 9 then MOB_5_R= NotAppl;
```

Tabulate Frequencies.

First, calculate frequency distributions on the short set and two extended set WALKING questions.

Mobility: Degree of difficulty walking or climbing steps

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13424	77.5	80.1	80.1
	Some difficulty	2165	12.5	12.9	93.0
	A lot of difficulty	792	4.6	4.7	97.7
	Cannot do at all	380	2.2	2.3	100.0
	Total	16761	96.7	100.0	
Missing		565	3.3		
Total		17326	100.0		

MOB_4_R: Diff walking 100 yards on level ground w/o aid or equipment

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13892	80.2	84.8	84.8
	Some difficulty	1369	7.9	8.4	93.2
	A lot of difficulty	491	2.8	3.0	96.2
	Cannot do at all	623	3.6	3.8	100.0
	Total	16375	94.5	100.0	
Missing		951	5.5		
Total		17326	100.0		

For more information on the Washington Group on Disability Statistics, visit:

<http://www.washingtongroup-disability.com/>.

MOB_5_R: Diff walking 1/3rd mile on level ground w/o aid or equipment

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	No difficulty	13025	75.2	82.8	82.8
	Some difficulty	1650	9.5	10.5	93.3
	A lot of difficulty	708	4.1	4.5	97.8
	Cannot do at all	347	2.0	2.2	100.0
	Total	15730	90.8	100.0	
Missing		1596	9.2		
Total		17326	100.0		

Step 20. Generate a cross-tabulation of the walking distance questions: MOB_4_R and MOB_5_R.

The syntax below produces a cross-tabulation of the two Extended Set WALKING questions: MOB_4_R: *Difficulty walking 100 yards without equipment* and MOB_5_R: *Difficulty walking 1/3 mile without equipment* to determine a single WALKING INDICATOR.

NOTE: **623** individuals who responded cannot do at all to MOB_4_R were not asked MOB_5_R and they do not appear in the table below. They are, however, accounted for in the WALKING indicator calculation.

Cross Tabulations.

MOB_4_R: Diff walking 100 yards on level ground w/o aid or equipment	MOB_5_R: Diff walking 1/3rd mile on level ground w/o aid or equipment				Total
	No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	
No difficulty	12950	819	63	39	13871
Some difficulty	72	810	343	142	1367
A lot of difficulty	3	21	301	166	491
Cannot do at all (623)	0	0	0	0	0
Total	13025	1650	707	347	15729

Step 21. Create a WALKING INDICATOR (WALK_INDICATOR) based on the two additional walking questions MOB_4_R and MOB_5_R.

Syntax below creates WALKING_INDICATOR based on the distribution in the cross-tabulation above.

```
PROC WALK_INDICATOR
WALK_INDICATOR=0;
if (MOB_4_R = 1 and (MOB_5_R = 1 or MOB_5_R = 2)) then WALK_INDICATOR = 1;
```

elseif (MOB_4_R = **1** and MOB_5_R = **3**) or (MOB_4_R = **2** and (MOB_5_R = **1** or MOB_5_R = **2** or MOB_5_R = **3**)) **then** WALK_INDICATOR = **2**;
elseif (MOB_4_R = **1** and MOB_5_R = **4**) or (MOB_4_R = **3** and (MOB_5_R = **1** or MOB_5_R = **2** or MOB_5_R = **3**)) **then** WALK_INDICATOR = **3**;
elseif (MOB_4_R = **2** and MOB_5_R = **4**) or (MOB_4_R = **3** and MOB_5_R = **4**) **then** WALK_INDICATOR = **4**;

Syntax below includes the **623** who responded cannot do at all to MOB_4_R into the WALKING INDICATOR.

elseif (WALK_INDICATOR = **0** AND MOB_4_R = **4**) **then** WALK_INDICATOR = **4**;
endif;
IF WALK_INDICATOR=**0** **then** WALK_INDICATOR=**NotAppl**;
endif;

Tabulate Frequencies.

		WALK_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	13769	79.5	84.2	84.2
	2.00	1288	7.4	7.9	92.1
	3.00	364	2.1	2.2	94.3
	4.00	931	5.4	5.7	100.0
	Total	16352	94.4	100.0	
Missing		974	5.6		
Total		17326	100.0		

Step 22. Supplement Walking Indicator with information on difficulty Climbing steps (MOB_6).

Syntax below adds information from MOB_6 on *difficulty climbing up or down 12 steps* to create a combined Mobility Indicator (MOB_INDICATOR).

If MOB_6 in **1, 2, 3, 4** **then** MOB_6_R=MOB_6;
ElseIf MOB_6 in **7, 8, 9** **then** MOB_6_R= **NotAppl**;
endif;

Cross Tabulations.

		MOB_6_R: Difficulty climbing up or down 12 steps				
WALK_INDICATOR		No difficulty	Some difficulty	A lot of difficulty	Cannot do at all	Total
walk_INDICATOR2	1.00	13048	645	55	17	13765
	2.00	370	767	135	16	1288
	3.00	43	106	200	14	363
	4.00	79	242	226	384	931
Total		13540	1760	616	431	16347

According to the table above, the syntax below reclassifies:

1. 55 individuals with level **1** on the WALKING INDICATOR as level **2** on the MOBILITY INDICATOR
2. 17 individuals with level **1** on the WALKING INDICATOR and 135 individuals with level **2** on the WALKING INDICATOR as level **3** on the MOBILITY INDICATOR, and
3. 16 individuals with level **2** on the WALKING INDICATOR as level **4** on the MOBILITY INDICATOR.

Step 23. Create a mobility indication (*MOB_INDICATOR*) with information garnered from cross-tabulation above.

```

PROC MOB_INDICATOR
MOB_INDICATOR = WALK_INDICATOR;
if (WALK_INDICATOR = 2 and MOB_6_R = 3) then MOB_INDICATOR = 3;
elseif (WALK_INDICATOR = 1 and MOB_6_R = 3) then MOB_INDICATOR = 2;
elseif (WALK_INDICATOR = 2 and MOB_6_R = 4) then MOB_INDICATOR = 4;
elseif (WALK_INDICATOR = 1 and MOB_6_R = 4) then MOB_INDICATOR = 3;
endif;

```

Tabulate Frequencies.

		MOB_IDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	13697	79.1	83.8	83.8
	2.00	1192	6.9	7.3	91.1
	3.00	516	3.0	3.2	94.2
	4.00	947	5.5	5.8	100.0
	Total	16352	94.4	100.0	
Missing		974	5.6		
Total		17326	100.0		

ANXIETY

Step 24. Generate frequency distribution on ANX_1.

First, calculate frequency distributions on ANX_1: *How often do you feel worried, nervous or anxious?*

```
if ANX_1 in 1, 2, 3, 4 then ANX_1_R = ANX_1;  
elseif ANX_1 in 7, 8, 9 then ANX_1_R = NotAppl;  
endif;
```

Tabulate Frequencies.

ANX_1_R: How often feel worried, nervous, or anxious?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Daily	1632	9.4	9.8	9.8
	Weekly	1872	10.8	11.2	21.0
	Monthly	1558	9.0	9.3	30.4
	A few times a year	4898	28.3	29.4	59.7
	Never	6714	38.8	40.3	100.0
	Total	16674	96.2	100.0	
Missing		652	3.8		
Total		17326	100.0		

Step 25. The syntax below recodes ANX_3 into ANX_3Y

- 1) to create a NOT ASKED category based on those who responded NEVER to ANX_1_R and
- 2) to place "SOMEWHERE BETWEEN" numerically in-between "A LITTLE" and "A LOT".

```
if ANX_3R = 1 then ANX_3Y = 1;  
elseif ANX_3R = 2 then ANX_3Y = 3;  
elseif ANX_3R = 3 then ANX_3Y = 2;  
elseif ANX_3R in 7, 8, 9 then ANX_3Y = NotAppl;  
endif;
```

Recode ANX_3Y to 0 (not asked) if ANX_1 is 5 (Never).

```
if ANX_1 = 5 then ANX_3Y = 0;  
endif;
```

Tabulate Frequencies.

ANX_3Y: Level of feelings last time felt worried/nervous/anxious

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	6714	38.8	40.3	40.3
	A little	5700	32.9	34.2	74.5
	In between a little and a lot	3076	17.8	18.5	92.9
	A lot	1176	6.8	7.1	100.0
	Total	16666	96.2	100.0	
Missing		660	3.8		
Total		17326	100.0		

Step 26. Generate a cross-tabulation of the anxiety Extended Set questions: ANX_1_R and ANX_3Y.

The syntax below produces a cross-tabulation of ANX_1_R: *How often you felt worried, nervous or anxious* (a measure of frequency) and ANX_3Y: *The level of those feeling the last time you felt worried, nervous or anxious* (a measure of intensity) – used to determine a single ANXIETY INDICATOR (ANX_INDICATOR).

Cross Tabulations.

ANX_3Y: Level of feelings last time felt worried, nervous or anxious	ANX_1_R: How often feel worried, nervous or anxious?					Total
	Daily	Weekly	Monthly	A Few Times A Year	Never	
Not asked	0	0	0	0	6714	6714
A little	489	887	897	3417	0	5690
In between a little and a lot	589	725	535	1221	0	3070
A lot	548	256	123	248	0	1175
Total	1626	1868	1555	4886	6714	16649

Step 27. Create an ANXIETY INDICATOR (ANX_INDICATOR) based on the two anxiety questions ANX_1_R and ANX_3Y.

Syntax below creates ANX_INDICATOR based on the distribution in the cross-tabulation above.

```
PROC ANX_INDICATOR
if ANX_1_R = NotAppl or ANX_3Y = NotAppl then ANX_INDICATOR = NotAppl;
elseif (ANX_3Y <= 4 and (ANX_1_R = 4 or ANX_1_R = 5)) then ANX_INDICATOR=1;
elseif ((ANX_1_R = 3) or (ANX_1_R < 3 and ANX_3Y=1) or (ANX_1_R = 2 and ANX_3Y = 2))
then ANX_INDICATOR = 2;
elseif ((ANX_1_R = 1 and ANX_3Y = 2) or (ANX_1_R = 2 and ANX_3Y = 3)) then
ANX_INDICATOR = 3;
```

elseif (ANX_1_R = **1** and ANX_3Y = **3**) **then** ANX_INDICATOR = **4**;
endif;

Tabulate Frequencies.

		ANX_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	11600	67.0	69.7	69.7
	2.00	3656	21.1	22.0	91.6
	3.00	845	4.9	5.1	96.7
	4.00	548	3.2	3.3	100.0
	Total	16649	96.1	100.0	
Missing		677	3.9		
Total		17326	100.0		

DEPRESSION

Step 28. Generate frequency distribution on DEP_1.

First, calculate frequency distributions on DEP_1: *How often do you feel depressed?*

PROC DEP_1_R

if DEP_1 **in** **1, 2, 3, 4, 5** **then** DEP_1_R = DEP_1;
elseif DEP_1 **in** **7, 8, 9** **then** DEP_1_R = **NotAppl**;
else DEP_1_R = **NotAppl**;
endif;

Tabulate Frequencies.

		DEP_1_R: How often do you feel depressed?			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Daily	756	4.4	4.5	4.5
	Weekly	926	5.3	5.6	10.1
	Monthly	1038	6.0	6.2	16.3
	A few times a year	4012	23.2	24.1	40.4
	Never	9929	57.3	59.6	100.0
	Total	16661	96.2	100.0	
Missing		665	3.8		
Total		17326	100.0		

Step 29. The syntax below recodes DEP_3 into DEP_3Y to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”. It also creates the category NOT ASKED, if DEP_1 is NEVER (1)

```
PROC DEP_3Y
if DEP_3R = 1 then DEP_3Y = 1;
elseif DEP_3R = 2 then DEP_3Y = 3;
elseif DEP_3R = 3 then DEP_3Y = 2;
elseif DEP_3R in 7, 8, 9 then DEP_3Y = NotAppl;
endif;
if DEP_1 =5 then DEP_3Y=0;
endif;
```

Tabulate Frequencies.

DEP_3Y: Level of feelings last time felt depressed

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	9929	57.3	59.6	59.6
	A little	3775	21.8	22.7	82.3
	In between a little and a lot	2016	11.6	12.1	94.4
	A lot	935	5.4	5.6	100.0
	Total	16655	96.1	100.0	
Missing		671	3.9		
Total		17326	100.0		

Step 30. Generate a cross-tabulation of the depression Extended Set questions: DEP_1_R and DEP_3Y.

The syntax below produces a cross-tabulation of DEP_1_R: *How often do you feel depressed* (a measure of frequency) and DEP_3Y: *The level of those feeling the last time you felt depressed* (a measure of intensity) – used to determine a single DEPRESSION INDICATOR (DEP_INDICATOR).

Cross Tabulations.

DEP_3Y: Level of feelings last time felt depressed	DEP_1_R: How often do you feel depressed?					Total
	Daily	Weekly	Monthly	A Few Times A Year	Never	
Not asked	0	0	0	0	9929	9929
A little	161	346	548	2708	0	3763
In between a little and a lot	209	384	378	1042	0	2013
A lot	381	191	112	248	0	932
Total	751	921	1038	3998	9929	16637

Step 31. Create a *DEPRESSION INDICATOR* (*DEP_INDICATOR*) based on the two depression questions *DEP_1_R* and *DEP_3Y*.

Syntax below creates *DEP_INDICATOR* based on the distribution in the cross-tabulation above.

```
PROC DEP_INDICATOR
if DEP_1_R = NotAppl or DEP_3Y = NotAppl then DEP_INDICATOR = NotAppl;
elseif (DEP_3Y <= 4 and (DEP_1_R = 4 or DEP_1_R = 5)) then DEP_INDICATOR = 1;
elseif ((DEP_1_R = 3) or (DEP_1_R < 3 and DEP_3Y=1) or (DEP_1 = 2 and DEP_3Y = 2)) then
DEP_INDICATOR = 2;
elseif ((DEP_1_R=1 and DEP_3Y=2) or (DEP_1_R=2 and DEP_3Y=3)) then DEP_INDICATOR = 3;
elseif (DEP_1_R = 1 and DEP_3Y = 3) then DEP_INDICATOR = 4;
endif;
```

Tabulate Frequencies.

		DEP_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	13927	80.4	83.7	83.7
	2.00	1929	11.1	11.6	95.3
	3.00	400	2.3	2.4	97.7
	4.00	381	2.2	2.3	100.0
	Total	16637	96.0	100.0	
Missing		689	4.0		
Total		17326	100.0		

PAIN

Step 32. Generate frequency distribution on *PAIN_2*.

First, calculate frequency distributions on *PAIN_2*: *Frequency of pain in the past 3 months*.

```
PROC PAIN_2_R
if PAIN_2 in 1, 2, 3, 4 then PAIN_2_R = PAIN_2;
elseif PAIN_2 in 7, 8, 9 then PAIN_2_R = NotAppl;
endif;
```

Tabulate Frequencies.

PAIN_2_R: Frequency of pain in past 3 months

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Never	6636	38.3	39.8	39.8
	Some days	6556	37.8	39.3	79.2
	Most days	1227	7.1	7.4	86.5
	Every day	2245	13.0	13.5	100.0
	Total	16664	96.2	100.0	
Missing		662	3.8		
Total		17326	100.0		

Step 33. The syntax below recodes PAIN_4 into PAIN_4Y to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”. It also creates the category NOT ASKED, if PAIN_2 is NEVER (1).

```
PROC PAIN_4Y
if PAIN_4 = 1 then PAIN_4Y = 1;
elseif PAIN_4 = 2 then PAIN_4Y = 3;
elseif PAIN_4 = 3 then PAIN_4Y = 2;
elseif PAIN_4 in 7, 8, 9 then PAIN_4Y = NotAppl;
endif;

if PAIN_2 = 1 then PAIN_4Y = 0;
endif;
```

Tabulate Frequencies.

PAIN_4Y: How much pain you had last time you had pain?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	6636	38.3	39.8	39.8
	A little	4865	28.1	29.2	69.0
	In between a little and a lot	3296	19.0	19.8	88.8
	A lot	1869	10.8	11.2	100.0
	Total	16666	96.2	100.0	
Missing		660	3.8		
Total		17326	100.0		

Step 34. Generate a cross-tabulation of the PAIN Extended Set questions: PAIN_2_R and PAIN_4Y.

The syntax below produces a cross-tabulation of PAIN_2_R: *Frequency of pain in the past 3 months* and PAIN_4Y: *How much pain you has the last time you had pain* (a measure of intensity) – used to determine a single PAIN INDICATOR (P_INDICATOR).

Cross Tabulations.

PAIN_4Y: How much pain you had last time you had pain		PAIN_2_R: Frequency of pain in past 3 months				
		Never	Some days	Most days	Every day	Total
Not asked		6636	0	0	0	6636
A little		0	4136	323	401	4860
In between a little and a lot		0	1772	624	896	3296
A lot		0	645	278	944	1867
Total		6636	6553	1225	2241	16655

Step 35. Create a PAIN INDICATOR (P_INDICATOR) based on the two PAIN questions PAIN_2_R and PAIN_4Y.

Syntax below creates P_INDICATOR based on the distribution in the cross-tabulation above.

```
PROC P_INDICATOR
  if (PAIN_2_R = 1) or (PAIN_4Y = 1 and (PAIN_2_R = 2 or PAIN_2_R = 3)) then
    P_INDICATOR = 1;
  elseif ((PAIN_2_R = 2 and (PAIN_4Y = 2 or PAIN_4Y = 3)) or (PAIN_2_R = 3 and
    PAIN_4Y = 2) or (PAIN_2_R = 4 and PAIN_4Y = 1)) then P_INDICATOR = 2;
  elseif (PAIN_2_R = 3 and PAIN_4Y = 3) or (PAIN_2_R = 4 and PAIN_4Y = 2)
  then P_INDICATOR = 3;
  elseif (PAIN_2_R = 4 and PAIN_4Y = 3) then P_INDICATOR = 4;
endif;
```

Tabulate Frequencies.

		P_INDICATOR			
		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	1.00	11095	64.0	66.6	66.6
	2.00	3442	19.9	20.7	87.3
	3.00	1174	6.8	7.0	94.3
	4.00	944	5.4	5.7	100.0
	Total	16655	96.1	100.0	
Missing		671	3.9		
Total		17326	100.0		

FATIGUE (Tired)

Step 36. Generate frequency distribution on FATIGUE Extended Set questions Tired_1, Tired_2 and Tired_3.

First, calculate frequency distributions on TIRED_1: How often you felt tired in the past 3 months.

```

PROC TIRED_1_R
if TIRED_1 in 1, 2, 3, 4 then TIRED_1_R = TIRED_1;
elseif TIRED_1 in 7, 8, 9 then TIRED_1_R = NotAppl;
endif;

```

Tabulate Frequencies. TIRED_1_R: How often felt very tired or exhausted in past 3 months

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Never	5619	32.4	33.7	33.7
	Some days	8391	48.4	50.4	84.1
	Most days	1632	9.4	9.8	93.9
	Every day	1019	5.9	6.1	100.0
	Total	16661	96.2	100.0	
Missing		665	3.8		
Total		17326	100.0		

Step 37. Recode Tired_2 to 0 (not asked) if Tired_1 is 1 (Never).

If response to TIRED_1 is 1: Never, then TIRED_2 (How long most recent tired or exhausted feelings lasted) is not asked. This variable is recoded so these individuals are included in the syntax below.

```

PROC TIRED_2_R
if TIRED_2 in 1, 2, 3 then TIRED_2_R = TIRED_2;
elseif TIRED_2 in 7, 8, 9 then TIRED_2_R = NotAppl;
else TIRED_2_R = NotAppl;
endif;

```

```

if TIRED_1 =1 then TIRED_2_R = 0;
endif;

```

Tabulate Frequencies.

TIRED_2_R: How long most recent tired or exhausted feelings lasted?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	5619	32.4	33.8	33.8
	Some of the day	8036	46.4	48.3	82.0
	Most of the day	1955	11.3	11.7	93.8
	All of the day	1036	6.0	6.2	100.0
	Total	16646	96.1	100.0	
Missing		680	3.9		
Total		17326	100.0		

Step 38. The syntax below recodes TIRE₃ into TIRE_{3Y} to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”.

Also, if response to TIRE₁ is 1: Never, then TIRE₃ (Level of tiredness) is not asked. This variable is recoded, so these individuals are included in the syntax below.

```
PROC TIRE3Y
if TIRE3 = 1 then TIRE3Y = 1;
elseif TIRE3 = 2 then TIRE3Y = 3;
elseif TIRE3 = 3 then TIRE3Y = 2;
elseif TIRE3 in 7, 8, 9 then TIRE3Y = NotAppl;
endif;

if TIRE1=1 then TIRE3Y=0;
endif;
```

Tabulate Frequencies.

TIRE_{3Y}: Level of tiredness

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not asked	5619	32.4	33.8	33.8
	A little	4912	28.4	29.5	63.3
	In between a little and a lot	4030	23.3	24.2	87.5
	A lot	2087	12.0	12.5	100.0
	Total	16648	96.1	100.0	
Missing		678	3.9		
Total		17326	100.0		

Step 39. Generate a cross-tabulation of the FATIGUE Extended Set questions: TIRE_{1_R}, TIRE_{2_R} and TIRE_{3Y}.

The syntax below produces a cross-tabulation of TIRE_{1_R}: *How often you felt tired or exhausted in the past 3 months* (a measure of frequency) and TIRE_{2_R}: *How long those feelings lasted* (a measure of duration) and TIRE_{3Y}: *The level of tiredness* (a measure of intensity) – used to determine a single TIRE_{3Y} INDICATOR (T_INDICATOR).

Cross Tabulations.

TIRED_3Y: Level of tiredness: <i>Intensity</i>	TIRED_2: How long feelings lasted: <i>Duration</i>	TIRED_1: How often felt very tired or exhausted in past 3 months: <i>Frequency</i>				Total
		Never	Some days	Most days	Every day	
Not asked	Not asked	5619				5619
A little	Some of the day		4066	264	124	4454
	Most of the day		252	73	27	352
	All of the day		68	15	18	101
In between	Some of the day		2224	400	166	2791
	Most of the day		497	266	123	887
	All of the day		194	71	84	349
A lot	Some of the day		536	165	84	785
	Most of the day		297	255	160	713
	All of the day		237	118	230	585
TOTAL		5619	8371	1627	1016	16633

Step 40. Create a FATIGUE INDICATOR (T_INDICATOR) based on the three FATIGUE questions TIRED_1_R, TIRED_2_R and TIRED_3Y.

Syntax below creates T_INDICATOR based on the distribution in the cross-tabulation above.

```

PROC T_INDICATOR
if TIRED_1_R = 1 then T_INDICATOR = 1;
elseif TIRED_1_R = 2 and TIRED_2_R = 1 and TIRED_3Y = 1 then T_INDICATOR = 1;
elseif TIRED_1_R = 3 and TIRED_2_R = 1 and TIRED_3Y = 1 then T_INDICATOR = 1;

elseif TIRED_1_R in 2, 3, 4 and TIRED_2_R in 2, 3 and TIRED_3Y = 1 then T_INDICATOR = 2;
elseif TIRED_1_R in 2, 3, 4 and TIRED_2_R = 1 and TIRED_3Y = 2 then T_INDICATOR = 2;
elseif TIRED_1_R = 2 and TIRED_2_R = 2 and TIRED_3Y = 2 then T_INDICATOR = 2;
elseif TIRED_1_R = 4 and TIRED_2_R = 1 and TIRED_3Y = 1 then T_INDICATOR = 2;
elseif TIRED_1_R in 3, 4 and TIRED_2_R = 2 and TIRED_3Y = 2 then T_INDICATOR = 3;
elseif TIRED_1_R in 2, 3, 4 and TIRED_2_R = 3 and TIRED_3Y = 2 then T_INDICATOR = 3;
elseif TIRED_1_R in 2, 3, 4 and TIRED_2_R = 1 and TIRED_3Y = 3 then T_INDICATOR = 3;
elseif TIRED_1_R in 2, 3 and TIRED_2_R = 2 and TIRED_3Y = 3 then T_INDICATOR = 3;
elseif TIRED_1_R = 2 and TIRED_2_R = 3 and TIRED_3Y = 3 then T_INDICATOR = 3;
elseif TIRED_1_R = 4 and TIRED_2_R = 2 and TIRED_3Y = 3 then T_INDICATOR = 4;
elseif TIRED_1_R in 3, 4 and TIRED_2_R = 3 and TIRED_3Y = 3 then T_INDICATOR = 4;
else T_INDICATOR = NotAppl;
endif;

```

Tabulate Frequencies.

		T_Indicator		Valid	Cumulative
		Frequency	Percent	Percent	Percent
Valid	1.00	9949	57.4	59.8	59.8
	2.00	3864	22.3	23.2	83.0
	3.00	2312	13.3	13.9	96.9
	4.00	508	2.9	3.1	100.0
	Total	16633	96.0	100.0	
Missing		693	4.0		
Total		17326	100.0		

Creating Disability Status Indicators

Type of Disability Indicator		Number of Questions
SS_1	Short Set (SS)	6
Extended Set		
ES_1	SS + <u>Hearing-indicator</u> , <u>Mobility-indicator</u> , <u>Cognition-indicator</u> , <u>Upper Body-indicator</u> + PFAD (4)*	25
ES_2	SS + <u>Hearing-indicator</u> , <u>Mobility-indicator</u> , <u>Cognition-indicator</u> , <u>Upper Body-indicator</u> + AD (4)†	20
Short Set Enhanced		
ES_3	SS + Upper Body-indicator + AD (4)†	12

* PFAD (4): Pain, Fatigue, Anxiety and Depression Indicators at level 4

† AD (4): Anxiety and Depression Indicators at level 4

SS_1: WG Short Set Disability Indicator based on the 6 short set questions

The syntax below calculates the WG Short Set Disability Indicator based on the six short set questions **SS_1** at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL.

PROC SS_1

if VISION = **NotAppl** and HEARING = **NotAppl** and MOBILITY = **NotAppl** and

COMMUNICATION = **NotAppl** and SELF_CARE = **NotAppl** and COGNITION = **NotAppl**

then SS_1 = **NotAppl**;


```

elseif VISION in 3, 4 or HEARING in 3, 4 or MOBILITY in 3, 4 or COMMUNICATION in 3,
4 or SELF_CARE in 3, 4 or COGNITION in 3, 4 then SS_1= 1;
else SS_1 = 2;
endif;

```

Tabulate Frequencies.

SS_1: WG Short Set Disability Identifier

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	14905	86.0	88.8	90.5
	WITH DISABILITY	1872	10.8	11.2	9.5
	Total	16777	96.8	100.0	
Missing		549	3.2		
Total		17326	100.0		

ES_1: SS_1 + Hearing-indicator, Mobility-indicator, Cognition-indicator + Upper Body-indicator + PFAD (4)

The syntax below calculates an Extended Set Disability Indicator (**ES_1**) based on 25 questions at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL for the six short set question; severity levels 3 or 4 for the Hearing-, Mobility-, Cognition- and Upper body-Indicators; and severity level 4 for Pain-, Fatigue-, Anxiety- and Depression-Indicators.

```

PROC ES_1
if ( SS_1 = NotAppl and (H_INDICATOR < 1 or H_INDICATOR > 4) and
(MOB_INDICATOR < 1 or MOB_INDICATOR > 4) and COM_SS = NotAppl and UB_SS =
NotAppl and (COG_INDICATOR < 1 or COG_INDICATOR > 4) and (UB_INDICATOR < 1
or UB_INDICATOR > 4) and P_INDICATOR = NotAppl and (T_INDICATOR < 1 or
T_INDICATOR > 4) and (ANX_INDICATOR < 1 or ANX_INDICATOR > 4) and
(DEP_INDICATOR < 1 or DEP_INDICATOR > 4)) then ES_1 = NotAppl;
elseif (SS_1 = 1 or (H_INDICATOR in 3, 4) or (MOB_INDICATOR in 3, 4) or
(COG_INDICATOR in 3, 4) or (UB_INDICATOR in 3, 4) or P_INDICATOR = 4 or
T_INDICATOR = 4 or ANX_INDICATOR = 4 or DEP_INDICATOR = 4) then ES_1 = 1;
else ES_1 = 2;
endif;

```

Tabulate Frequencies.

ES_1: WG-ES Disability Indicator based on 10 domains and 25 questions

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	13823	79.8	79.8	82.3
	WITH DISABILITY	3503	20.2	20.2	17.7
	Total	17326	100.0	100.0	

ES_2: SS_1 + Hearing-indicator, Mobility-indicator, Cognition-indicator + Upper Body-indicator + AD (4)

The syntax below calculates an Extended Set Disability Indicator (**ES_2**) based on 20 questions at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded A LOT OF DIFFICULTY or CANNOT DO AT ALL for the six short set question; severity levels 3 or 4 for the Hearing-, Mobility-, Cognition- and Upper body-Indicators; and severity level 4 for Anxiety- and Depression-Indicators.

PROC ES_2

```

if (SS_1 = NotAppl and (H_INDICATOR < 1 or H_INDICATOR > 4) and (MOB_INDICATOR < 1
or MOB_INDICATOR > 4) and COM_SS = NotAppl and UB_SS = NotAppl and
(COG_INDICATOR < 1 or COG_INDICATOR > 4) and (UB_INDICATOR < 1 or UB_INDICATOR
> 4) and (ANX_INDICATOR < 1 or ANX_INDICATOR > 4) and (DEP_INDICATOR < 1 or
DEP_INDICATOR > 4)) then ES_2 = NotAppl;
elseif (SS_1 = 1 or (H_INDICATOR = 3 or H_INDICATOR = 4) or (MOB_INDICATOR = 3 or
MOB_INDICATOR = 4) or (COG_INDICATOR = 3 or COG_INDICATOR = 4) or
(UB_INDICATOR = 3 or UB_INDICATOR = 4) or ANX_INDICATOR = 4 or DEP_INDICATOR =
4) then ES_2 = 1;
else ES_2 = 2;
endif;

```

Tabulate Frequencies.

ES_2: WG-ES Disability Indicator based on 8 domains and 20 questions

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	14222	82.1	82.1	84.6
	WITH DISABILITY	3104	17.9	17.9	15.4
	Total	17326	100.0	100.0	

ES_3: SS_1 + Upper Body-indicator + AD (4)

The syntax below calculates the WG Short Set ENHANCED Disability Indicator (**ES_3**) based on the 12 questions at the recommended cut-off. The level of inclusion is: at least 1 domain/question is coded

A LOT OF DIFFICULTY or CANNOT DO AT ALL for the six short set question; severity levels 3 or 4 for the Upper body-Indicators; and severity level 4 for Anxiety- and Depression-Indicators.

PROC ES_3

```

IF VISION = NotAppl and HEARING = NotAppl and MOBILITY = NotAppl and
COMMUNICATION = NotAppl and SELF_CARE = NotAppl and COGNITION = NotAppl
and UB_INDICATOR = NotAppl and ANX_INDICATOR = NotAppl and DEP_INDICATOR
= NotAppl then ES_3 = NotAppl;
elseif VISION in 3, 4 or HEARING in 3, 4 or MOBILITY in 3, 4 or COMMUNICATION in
3, 4 or SELF_CARE in 3, 4 or COGNITION in 3, 4 or UB_INDICATOR in 3, 4 or
ANX_INDICATOR = 4 or DEP_INDICATOR = 4 then ES_3 = 1;
else ES_3 = 2;
endif;

```

Tabulate Frequencies.

ES_3: WG-SS Enhanced Disability Indicator based on 8 domains and 12 questions

		Frequency	Percent	Valid Percent	Weighted Estimate
Valid	WITHOUT DISABILITY	14393	83.1	85.8	87.7
	WITH DISABILITY	2384	13.8	14.2	12.3
	Total	16777	96.9	100.0	
Missing		549	3.2		
Total		17326	100.1		

POSTSCRIPT: Why Exclude Pain and Fatigue?

Of note is the exclusion of the pain and fatigue domains from several of the Disability Identifiers above. There has been considerable discussion within the WG on these domains. They are not, strictly speaking, domains of functioning – and our analyses indicated that they are both highly correlated with other domains – and that the rates of disability with the inclusion of these domains can be very high. Finally, in terms of international comparability, these domains are less universal; that is, they are more susceptible to local, socio-cultural influences than other domains of functioning.

For these reasons, we chose to exclude them from several of these analyses, though they can be included in supplemental analyses carried out by NSOs on a national basis.

APPENDIX

CSPRO Code Used with the 2013 NHIS Data File

PROC GLOBAL

PROC NIHS2013_FUNCTIONING_AND_DISA_FF

PROC VISION

// *Step 1. Generate frequency distribution for each domain question: Vision

//VIS_SS is the WG-SS Vision question;

//*****;

//*Convert 7,8,9 to missing;

PROC VISION

If VIS_SS **in** 1, 2, 3, 4 **then** Vision=VIS_SS;

ElseIf VIS_SS **in** 7, 8, 9 **then** Vision=**NotAppl**;

EndIf;

// *Step 2. Generate frequency distribution for each domain question: Communication

// COM_SS is the WG-SS Communication question;

//*****;

//*Convert 7,8,9 to missing;

PROC COMMUNICATION

If COM_SS **in** 1, 2, 3,4 **then** Communication =COM_SS;

ElseIf COM_SS **in** 7, 8, 9 **then** Communication = **NotAppl**;

EndIf;

// *Step 3. Generate frequency distribution for each domain question (including domains with multiple questions): Hearing

// HEAR_SS is the WG-SS Hearing question.

// HEAR_3 is Difficulty hearing conversation with one person in quiet room.

// HEAR_4 is Difficulty hearing one person in noisier room;

//*****;

//*Convert 7,8,9 to;

PROC HEARING

If HEAR_SS **in** 1, 2, 3, 4 **then** Hearing =HEAR_SS;

ElseIf HEAR_SS **in** 7, 8, 9 **then** Hearing = **NotAppl**;

EndIf;

if HEAR_3 **in** 1, 2, 3, 4 **then** HEAR_3_R=HEAR_3;

elseif HEAR_3 **in** 7, 8, 9 **then** HEAR_3_R=**NotAppl**;

endif;

if HEAR_4 **in** 1, 2, 3, 4 **then** HEAR_4_R=HEAR_4;

elseif HEAR_4 **in** 7, 8, 9 **then** HEAR_4_R=**NotAppl**;

endif;

/Step 4. For Hearing questions, recode Hear_3_R and Hear_4_R to 4 (cannot do at all) if HEAR_SS2 is 4 (Cannot do at all). If HEAR_SS2 = 4 HEAR_3_R = 4. If HEAR_SS2 = 4 HEAR_4_R = 4.**

```
PROC HEAR_3_R  
if HEARING = 4 and HEAR_3_R = NotAppl then HEAR_3_X = 4;  
else HEAR_3_X = HEAR_3_R;  
endif;
```

```
PROC HEAR_4_R  
if HEARING = 4 and HEAR_4_R = NotAppl then HEAR_4_X = 4;  
else HEAR_4_X = HEAR_4_R;  
endif;
```

/Step 5. Generate a cross-tabulation of the two Hearing Extended Set questions: HEAR_3_X and HEAR_4_X.;**

/Step 6. Create a HEARING INDICATOR (H_INDICATOR) based on the two additional hearing questions HEAR_3_X and HEAR_4_X**

// The syntax below creates a HEARING INDICATOR (H_INDICATOR) based on the two additional hearing questions HEAR_3_X and HEAR_4_X;

```
PROC H_INDICATOR  
if (HEAR_3_X = 1 and HEAR_4_X = 1) or (HEAR_3_X = 1 and HEAR_4_X = 2)  
                                then H_INDICATOR = 1;  
elseif (HEAR_3_X = 2 and (HEAR_4_X = 1 or HEAR_4_X = 2)) or (HEAR_3_X = 1 and  
HEAR_4_X = 3) then H_INDICATOR = 2;  
elseif (HEAR_3_X = 3 and (HEAR_4_X = 1 or HEAR_4_X = 2) or (HEAR_3_X = 2 and  
HEAR_4_X = 3) or (HEAR_3_X = 1 and HEAR_4_X = 4)) then H_INDICATOR = 3;  
elseif ((HEAR_3_X = 3 and HEAR_4_X = 3) or HEAR_3_X = 4 or (HEAR_4_X = 4 and  
(HEAR_3_X = 2 or HEAR_3_X = 3))) then H_INDICATOR = 4;  
endif;
```

*******;**

/Step 7. Generate frequency distribution for each domain question: Cognition**

/COG_SS is the WG-SS Cognition question.**

/Cognition:Degree of difficulty remembering or concentrating;**

*******;**

/Recode 7,8,9 to NotAppl;**

```
PROC COGNITION  
if COG_SS in 1, 2, 3, 4 then COGNITION = COG_SS;  
elseif COG_SS in 7, 8, 9 then COGNITION = NotAppl;  
endif;
```

/If response to COG-SS 2, 3 or 4 then the respondent is asked COG_1: whether they have difficulty remembering, concentrating or both. ;**

```
PROC COG_1_R  
if COG_1 in 1, 2, 3 then COG_1_R = COG_1;  
elseif COG_1 in 7, 8, 9 then COG_1_R = NotAppl;  
endif;
```

// *Step 8. Account for those who did not answer COG_1 (COG_SS is 1 – no difficulty and they were skipped) by recoding COG_1 to 0 (No difficulty).
// If response to COG-SS is 1: no difficulty, then the variable COG_1 is recoded into COG_1A, and the value assigned is 0: no difficulty;

```
PROC COG_1A  
if COG_SS = 1 then COG_1A = 0;  
else COG_1A = COG_1_R;  
endif;
```

/**Step 9. Generate frequency distribution for remaining cognition questions.
// Frequency distribution of the Cognition extended REMEMBERING questions: COG_2 and COG_3;

```
PROC COG_2_R  
if COG_2 in 1,2,3 then COG_2_R=COG_2;  
elseif COG_2 in 7,8,9 then COG_2_R=NotAppl;  
endif;
```

```
PROC COG_3_R  
if COG_3 in 1,2,3 then COG_3_R = COG_3;  
elseif COG_3 in 7,8,9 then COG_3_R=NotAppl;  
endif;
```

/**Step 10. Generate cross-tabulation of the two Cognition extended set questions COG_2 by COG_3;

/**Step 11. Create a Remembering Indicator based on distribution of COG_2 and COG_3. The syntax below creates a REMEMBERING INDICATOR (R_INDICATOR)

//based on the additional remembering questions (COG_2_R and COG_3_R).

//If COG_SS is 1: no difficulty, then the Remembering Indicator is coded as 1: the lowest level of difficulty;

```
PROC R_INDICATOR  
if COGNITION = 1 then R_INDICATOR = 1;  
elseif ((COG_2_R = 1 and COG_3_R = 1) or (COG_3_R = 1 and COG_2_R = 2) or  
(COG_3_R = 2 and COG_2_R = 1)) then R_INDICATOR = 2;  
elseif (COG_3_R = 2 and COG_2_R = 2) then R_INDICATOR = 3;  
elseif (COG_3_R = 3 or COG_2_R = 3) then R_INDICATOR = 4;
```

/**Step 12. If COG_1A is coded as 2 (concentrating only), then the Remembering Indicator is coded as 5.

//These 388 individuals are respondents who were not included in the Remembering Indicator since they

//had only difficulty concentrating. ;

```
elseif (COG_1A = 2) then R_INDICATOR = 5;  
else R_INDICATOR = 0;  
endif;
```

/**Step 13. Generate frequency distribution of the Remembering Indicator;

```

/*Step 14. Supplement Remembering Indicator with information on difficulty concentrating.
//The syntax below adds information on whether cognitive difficulties are compounded by
difficulty
//concentrating in addition to difficulty remembering.
//Create a COGNITION INDICATOR (COG_INDICATOR) based on R_INDICATOR (above)
and the
//cognition question (COG_1). The 388 individuals with ‘concentrating only’ were allocated as
follows:
//1. 357 with a little difficulty on COG_SS question were classified as 2
//2. 30 with a lot of difficulty on COG_SS question were classified as 3
//3. 1 with cannot do on COG_SS question was classified as 4
//Those with both remembering and concentrating difficulty were upgraded 36 individuals from 2
to 3,
//and 125 individuals from 3 to 4.;
PROC COG_INDICATOR
COG_INDICATOR = R_INDICATOR;

if R_INDICATOR = 5 and COGNITION = 2 then COG_INDICATOR = 2;
elseif R_INDICATOR = 5 and COGNITION = 3 then COG_INDICATOR = 3;
elseif R_INDICATOR = 5 and Cognition = 4 then COG_INDICATOR = 4;
endif;

if R_INDICATOR = 2 and COG_1_R = 3 and Cognition = 3 then COG_INDICATOR = 3;
elseif R_INDICATOR=3 and COG_1_R=3 and COGNITION=3 then COG_INDICATOR = 4;
endif;
/*Step 15. Generate frequency distribution of the Cognition Indicator.;
*****;
*****;
/*Step 16. Generate frequency distribution for each domain question: Self-care and Upper body
functioning.;
/*UB_SS is the WG-SS Self-care question.
//Recode 7,8,9 to .;
PROC Self_Care
if UB_SS in 1, 2, 3, 4 then Self_Care = UB_SS;
elseif UB_SS in 7,8,9 then Self_Care =NotAppl;
endif;

PROC UB_1_R
if UB_1 in 1, 2, 3, 4 then UB_1_R = UB_1;
elseif UB_1 in 7,8,9 then UB_1_R=NotAppl;
endif;

PROC UB_2_R
if UB_2 in 1, 2, 3, 4 then UB_2_R = UB_2;
elseif UB_2 in 7,8,9 then UB_2_R=NotAppl;
endif;

```

/Step 17. Generate a cross-tabulation of the two Upper body Extended Set questions: UB_2_R and UB_1_R;**

/*****;**

/Step 18. Create an UPPER BODY INDICATOR (UB_INDICATOR) based on the two additional self care questions UB_2_R and UB_3_R.**

// Syntax below creates UB_INDICATOR based on the distribution in the cross-tabulation above;

```
PROC UB_INDICATOR
if UB_1_R = 4 or UB_2_R = 4 then UB_INDICATOR = 4;
elseif UB_INDICATOR <> 4 and (UB_1_R = 3 or UB_2_R = 3)
then UB_INDICATOR = 3;
elseif UB_INDICATOR <> 4 and UB_INDICATOR <> 3 and (UB_1_R = 2 or UB_2_R = 2)
then UB_INDICATOR = 2;
elseif UB_INDICATOR <> 4 and UB_INDICATOR <> 3 and UB_INDICATOR <> 2 and
(UB_1_R = 1 or UB_2_R = 1) then UB_INDICATOR = 1;
endif;
```

/*****;**

/*****;**

/Step 19. Generate frequency distribution for each domain question: Mobility.**

// MOB_SS is the WG-SS Mobility question

// First, calculate frequency distributions on the short set and two extended set WALKING questions (MOB_4,MOB_5);

```
PROC MOBILITY
if MOB_SS2 in 1, 2, 3, 4 then MOBILITY = MOB_SS2;
elseif MOB_SS2 in 7,8,9 then MOBILITY = NotAppl;
endif;
```

```
if MOB_4 in 1, 2, 3, 4 then MOB_4_R = MOB_4;
elseif MOB_4 in 7,8,9 then MOB_4_R=NotAppl;
endif;
```

```
if MOB_5 in 1, 2, 3, 4 then MOB_5_R = MOB_5;
elseif MOB_5 in 7,8,9 then MOB_5_R = NotAppl;
endif;
```

/Step 20. Generate a cross-tabulation of the walking distance questions: MOB_4_R (Difficulty walking 100 yards without equipment)**

//and MOB_5_R (Difficulty walking 1/3 mile without equipment).to determine a single WALKING INDICATOR.

// NOTE: 623 individuals who responded cannot do at all to MOB_4 were not asked MOB_5 and they do not appear in the table below.

//They are, however, accounted for in the WALKING indicator calculation;

/Step 21. Create a WALKING INDICATOR (WALK_INDICATOR) based on the two additional walking questions MOB_4_R and MOB_5_R.**

//Syntax below creates WALKING_INDICATOR based on the distribution in the cross-tabulation above.;

```
PROC WALK_INDICATOR
WALK_INDICATOR=0;
if (MOB_4_R = 1 and (MOB_5_R = 1 or MOB_5_R = 2)) then WALK_INDICATOR = 1;
elseif (MOB_4_R = 1 and MOB_5_R = 3) or (MOB_4_R = 2 and (MOB_5_R = 1 or
MOB_5_R = 2 or MOB_5_R = 3)) then WALK_INDICATOR = 2;
elseif (MOB_4_R = 1 and MOB_5_R = 4) or (MOB_4_R = 3 and (MOB_5_R = 1 or
MOB_5_R = 2 or MOB_5_R = 3)) then WALK_INDICATOR = 3;
elseif (MOB_4_R = 2 and MOB_5_R = 4) or (MOB_4_R = 3 and MOB_5_R = 4)
then WALK_INDICATOR = 4;
```

//Syntax below includes the 623 who responded cannot do at all to MOB_4 into the WALKING INDICATOR;

```
elseif (WALK_INDICATOR = 0 AND MOB_4_R = 4) then WALK_INDICATOR = 4;
endif;
```

//RECODE WALK_INDICATOR (0 = SYSMIS).;

```
IF WALK_INDICATOR=0 then WALK_INDICATOR=NotAppl;
endif;
```

/*Step 22. Supplement Walking Indicator with information on difficulty Climbing steps (MOB_6). CROSSTABS WALK_INDICATOR BY MOB_6.;

```
PROC MOB_6_R
if MOB_6 in 1, 2, 3, 4 then MOB_6_R = MOB_6;
elseif MOB_6 in 7,8,9 then MOB_6_R = NotAppl;
endif;
```

/*Step 23. Create a mobility indication (MOB_INDICATOR) with information garnered from cross-tabulation above.;

```
PROC MOB_INDICATOR
MOB_INDICATOR = WALK_INDICATOR;
if (WALK_INDICATOR = 2 and MOB_6_R = 3) then MOB_INDICATOR = 3;
elseif (WALK_INDICATOR = 1 and MOB_6_R = 3) then MOB_INDICATOR = 2;
elseif (WALK_INDICATOR = 2 and MOB_6_R = 4) then MOB_INDICATOR = 4;
elseif (WALK_INDICATOR = 1 and MOB_6_R = 4) then MOB_INDICATOR = 3;
endif;
```

*******;**

*******;**

/*Step 24. Generate frequency distribution on ANX_1 (anxiety);

```
PROC ANX_1_R
if ANX_1 in 1, 2, 3, 4 then ANX_1_R = ANX_1;
elseif ANX_1 in 7,8,9 then ANX_1_R=NotAppl;
endif;
```

/*Step 25. The syntax below recodes ANX_3R into ANX_3Y to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”;

```
PROC ANX_3Y
```

```

if ANX_3R = 1 then ANX_3Y = 1;
elseif ANX_3R = 2 then ANX_3Y = 3;
elseif ANX_3R = 3 then ANX_3Y = 2;
elseif ANX_3R in 7,8,9 then ANX_3Y = NotAppl;
endif;

```

/Recode ANX_3Y to 0 (not asked) if ANX_1 is 5 (Never).;**

```

if ANX_1 = 5 then ANX_3Y=0;
endif;

```

/Step 26. Generate a cross-tabulation of the anxiety Extended Set questions: ANX_1_R and ANX_3Y.;**

/Step 27. Create an ANXIETY INDICATOR (ANX_INDICATOR) based on the two anxiety questions ANX_1_R and ANX_3Y.**

/Syntax below creates ANX_INDICATOR based on the distribution in the cross-tabulation above.;**

```

PROC ANX_INDICATOR
if ANX_1_R = NotAppl or ANX_3Y = NotAppl then ANX_INDICATOR = NotAppl;
elseif (ANX_3Y <= 4 and (ANX_1_R = 4 or ANX_1_R = 5)) then ANX_INDICATOR=1;
elseif ((ANX_1_R = 3) or (ANX_1_R < 3 and ANX_3Y=1) or (ANX_1_R = 2 and ANX_3Y = 2)) then ANX_INDICATOR = 2;
elseif ((ANX_1_R = 1 and ANX_3Y = 2) or (ANX_1_R = 2 and ANX_3Y = 3)) then ANX_INDICATOR = 3;
elseif ( ANX_1_R = 1 and ANX_3Y = 3) then ANX_INDICATOR = 4;
endif;

```

/*****;**

/Step 28. Generate frequency distribution on DEP_1 (depression);**

```

PROC DEP_1_R
if DEP_1 in 1, 2, 3, 4, 5 then DEP_1_R = DEP_1;
elseif DEP_1 in 7,8,9 then DEP_1_R = NotAppl;
else DEP_1_R = NotAppl;
endif;

```

/Step 29. The syntax below recodes DEP_3R into DEP_3Y to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”;**

```

PROC DEP_3Y
if DEP_3R = 1 then DEP_3Y = 1;
elseif DEP_3R = 2 then DEP_3Y = 3;
elseif DEP_3R = 3 then DEP_3Y = 2;
elseif DEP_3R in 7,8,9 then DEP_3Y = NotAppl;
endif;

```

/Recode DEP_3Y to 0 (not asked) if DEP_1 is 5 (Never).;**

```

if DEP_1 = 5 then DEP_3Y=0;
endif;

```

/Step 30. Generate a cross-tabulation of the depression Extended Set questions: DEP_1_R and DEP_3Y.;**

/Step 31. Create a DEPRESSION INDICATOR (DEP_INDICATOR) based on the two depression questions DEP_1_R and DEP_3Y;**

```
PROC DEP_INDICATOR
if DEP_1_R = NotAppl or DEP_3Y = NotAppl then DEP_INDICATOR = NotAppl;
elseif (DEP_3Y <= 4 and (DEP_1_R = 4 or DEP_1_R = 5)) then DEP_INDICATOR = 1;
elseif ((DEP_1_R = 3) or (DEP_1_R < 3 and DEP_3Y=1) or (DEP_1 = 2 and DEP_3Y = 2))
then DEP_INDICATOR = 2;
elseif ((DEP_1_R=1 and DEP_3Y = 2) or (DEP_1_R = 2 and DEP_3Y = 3)) then
DEP_INDICATOR = 3;
elseif (DEP_1_R = 1 and DEP_3Y = 3) then DEP_INDICATOR = 4;
endif;
```

/*****;**

/Step 32. Generate frequency distribution on PAIN_2 (Pain);**

```
PROC PAIN_2_R
if PAIN_2 in 1, 2, 3, 4 then PAIN_2_R = PAIN_2;
elseif PAIN_2 in 7,8,9 then PAIN_2_R = NotAppl;
endif;
```

/Step 33. The syntax below recodes PAIN_4 into PAIN_4Y to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”**

// It also creates the category NOT ASKED, if PAIN_2 is NEVER (1);

```
PROC PAIN_4Y
if PAIN_4 = 1 then PAIN_4Y = 1;
elseif PAIN_4 = 2 then PAIN_4Y = 3;
elseif PAIN_4 = 3 then PAIN_4Y = 2;
elseif PAIN_4 in 7,8,9 then PAIN_4Y = NotAppl;
endif;
```

```
if PAIN_2 = 1 then PAIN_4Y = 0;
endif;
```

/Step 34. Generate a cross-tabulation of the PAIN Extended Set questions: PAIN_2_R and PAIN_4Y.;**

/Step 35. Create a PAIN INDICATOR (P_INDICATOR) based on the two PAIN questions PAIN_2_R and PAIN_4Y.**

//Syntax below creates P_INDICATOR based on the distribution in the cross-tabulation above.;

```
PROC P_INDICATOR
```

```
if (PAIN_2_R=1) or (PAIN_4Y= 1 and (PAIN_2_R = 2 or PAIN_2_R = 3)) then P_INDICATOR = 1;
elseif ((PAIN_2_R = 2 and (PAIN_4Y = 2 or PAIN_4Y = 3)) or (PAIN_2_R = 3 and PAIN_4Y = 2)
or (PAIN_2_R = 4 and PAIN_4Y = 1)) then P_INDICATOR = 2;
elseif (PAIN_2_R=3 and PAIN_4Y =3) or (PAIN_2_R=4 and PAIN_4Y=2) then P_INDICATOR = 3;
elseif (PAIN_2_R = 4 and PAIN_4Y = 3) then P_INDICATOR = 4;
endif;
```

```

*****;
/*Step 36. Generate frequency distribution on FATIGUE Extended Set questions Tired_1 (How often you felt tired in the past 3 months.), Tired_2 and Tired_3;
  PROC TIRED_1_R
  if TIRED_1 in 1, 2, 3, 4 then TIRED_1_R = TIRED_1;
  elseif TIRED_1 in 7,8,9 then TIRED_1_R = NotAppl;
  endif;

  PROC TIRED_2_R
  if TIRED_2 in 1, 2, 3 then TIRED_2_R = TIRED_2;
  elseif TIRED_2 in 7,8,9 then TIRED_2_R = NotAppl;
  else TIRED_2_R = NotAppl;
  endif;

/*Step 37. Recode Tired_2 to 0 (not asked) if Tired_1 is 1 (Never);
  if TIRED_1 =1 then TIRED_2_R = 0;
  endif;

/*Step 38. recodes TIRED_3 into TIRED_3Y to place “SOMEWHERE BETWEEN” numerically in-between “A LITTLE” and “A LOT”.
/* Also, if response to TIRED_1 is 1: Never, then TIRED_3 is not asked;
  PROC TIRED_3Y
  if TIRED_3 = 1 then TIRED_3Y = 1;
  elseif TIRED_3 = 2 then TIRED_3Y = 3;
  elseif TIRED_3 = 3 then TIRED_3Y = 2;
  elseif TIRED_3 in 7,8,9 then TIRED_3Y = NotAppl;
  endif;

  if TIRED_1=1 then TIRED_3Y=0;
  endif;

/*Step 39. Generate a cross-tabulation of the FATIGUE Extended Set questions: TIRED_1_R, TIRED_2_R and TIRED_3Y.;

/*Step 40. Create a FATIGUE INDICATOR (T_INDICATOR) based on the three FATIGUE questions TIRED_1_R, TIRED_2_R and TIRED_3Y.;
  PROC T_INDICATOR
  if TIRED_1_R = 1 then T_INDICATOR = 1;
  elseif TIRED_1_R = 2 and TIRED_2_R = 1 and TIRED_3Y = 1 then T_INDICATOR = 1;
  elseif TIRED_1_R = 3 and TIRED_2_R = 1 and TIRED_3Y = 1 then T_INDICATOR = 1;

  elseif TIRED_1_R in 2,3,4 and TIRED_2_R in 2,3 and TIRED_3Y=1 then T_INDICATOR= 2;
  elseif TIRED_1_R in 2,3,4 and TIRED_2_R = 1 and TIRED_3Y = 2 then T_INDICATOR = 2;
  elseif TIRED_1_R =2 and TIRED_2_R =2 and TIRED_3Y = 2 then T_INDICATOR = 2;
  elseif TIRED_1_R =4 and TIRED_2_R =1 and TIRED_3Y = 1 then T_INDICATOR = 2;
  elseif TIRED_1_R in 3,4 and TIRED_2_R =2 and TIRED_3Y = 2 then T_INDICATOR = 3;
  elseif TIRED_1_R in 2,3,4 and TIRED_2_R =3 and TIRED_3Y = 2 then T_INDICATOR = 3;
  elseif TIRED_1_R in 2,3,4 and TIRED_2_R =1 and TIRED_3Y = 3 then T_INDICATOR = 3;

```

```

elseif TIRED_1_R in 2,3 and TIRED_2_R = 2 and TIRED_3Y = 3 then T_INDICATOR = 3;
elseif TIRED_1_R = 2 and TIRED_2_R = 3 and TIRED_3Y = 3 then T_INDICATOR = 3;
elseif TIRED_1_R = 4 and TIRED_2_R = 2 and TIRED_3Y = 3 then T_INDICATOR = 4;
elseif TIRED_1_R in 3,4 and TIRED_2_R = 3 and TIRED_3Y = 3 then T_INDICATOR = 4;
else T_INDICATOR = NotAppl;
endif;

```

/SS_1: WG Short Set Disability Indicator based on the 6 short set questions.;**

```

PROC SS_1
if VISION = NotAppl and HEARING = NotAppl and MOBILITY = NotAppl and
COMMUNICATION = NotAppl and SELF_CARE = NotAppl and COGNITION = NotAppl
then SS_1 = NotAppl;
elseif VISION in 3, 4 or HEARING in 3, 4 or MOBILITY in 3, 4 or COMMUNICATION in 3,
4 or SELF_CARE in 3, 4 or COGNITION in 3, 4 then SS_1 = 1;
else SS_1 = 2;
endif;

```

/ES_1: SS_1 + Hearing-indicator, Mobility-indicator, Cognition-indicator + Upper Body-indicator + PFAD (4).**

//PFAD=only level 4 in Pain indictor, Fatigue indicator, Anxiety indictor, and Depression indicator;

```

PROC ES_1
if ( SS_1 = NotAppl and (H_INDICATOR < 1 or H_INDICATOR > 4) and
(MOB_INDICATOR < 1 or MOB_INDICATOR > 4) and COM_SS = NotAppl and UB_SS =
NotAppl and (COG_INDICATOR < 1 or COG_INDICATOR > 4) and (UB_INDICATOR < 1
or UB_INDICATOR > 4) and P_INDICATOR = NotAppl and (T_INDICATOR < 1 or
T_INDICATOR > 4) and (ANX_INDICATOR < 1 or ANX_INDICATOR > 4) and
(DEP_INDICATOR < 1 or DEP_INDICATOR > 4)) then ES_1 = NotAppl;
elseif (SS_1 = 1 or (H_INDICATOR in 3, 4) or (MOB_INDICATOR in 3, 4) or
(COG_INDICATOR in 3, 4) or (UB_INDICATOR in 3, 4) or P_INDICATOR = 4 or
T_INDICATOR = 4 or ANX_INDICATOR = 4 or DEP_INDICATOR = 4) then ES_1 = 1;
else ES_1 = 2;
endif;

```

/ES_2: SS_1 + Hearing-indicator, Mobility-indicator, Cognition-indicator + Upper Body-indicator + AD (4)**

// AD=only level 4 in Anxiety indictor and Depression indicator;

```

PROC ES_2
if (SS_1 = NotAppl and (H_INDICATOR < 1 or H_INDICATOR > 4) and
(MOB_INDICATOR < 1 or MOB_INDICATOR > 4) and COM_SS = NotAppl and UB_SS =
NotAppl and (COG_INDICATOR < 1 or COG_INDICATOR > 4) and (UB_INDICATOR < 1
or UB_INDICATOR > 4) and (ANX_INDICATOR < 1 or ANX_INDICATOR > 4) and
(DEP_INDICATOR < 1 or DEP_INDICATOR > 4)) then ES_2 = NotAppl;
elseif (SS_1 = 1 or (H_INDICATOR = 3 or H_INDICATOR = 4) or (MOB_INDICATOR = 3
or MOB_INDICATOR = 4) or (COG_INDICATOR = 3 or COG_INDICATOR = 4) or

```

```
(UB_INDICATOR = 3 or UB_INDICATOR = 4) or ANX_INDICATOR = 4 or  
DEP_INDICATOR = 4) then ES_2 = 1;  
else ES_2 = 2;  
endif;
```

```
/*ES_3: SS_1 + Upper Body-indicator + AD (4)
```

```
//AD=only level 4 in Anxiety indicator and Depression indicator;
```

```
PROC ES_3
```

```
IF VISION = NotAppl and HEARING = NotAppl and MOBILITY = NotAppl and  
COMMUNICATION = NotAppl and SELF_CARE = NotAppl and COGNITION = NotAppl  
and UB_INDICATOR = NotAppl and ANX_INDICATOR = NotAppl and DEP_INDICATOR  
= NotAppl then ES_3 = NotAppl;
```

```
elseif VISION in 3, 4 or HEARING in 3, 4 or MOBILITY in 3, 4 or COMMUNICATION in  
3, 4 or SELF_CARE in 3, 4 or COGNITION in 3, 4 or UB_INDICATOR in 3, 4 or  
ANX_INDICATOR = 4 or DEP_INDICATOR = 4 then ES_3 = 1;
```

```
else ES_3 = 2;
```

```
endif;
```